# **Documentación simple de Web Befine**

Ivan Rodríguez Pinares

Contenido

[Documentación simple de Web Befine 1](#_Toc203585458)

[🧱 Estructura de apps Django para Befine Web 3](#_Toc203585459)

[📦 Detalle de cada app 3](#_Toc203585460)

[1. core 3](#_Toc203585461)

[2. productos 3](#_Toc203585462)

[3. clientes 4](#_Toc203585463)

[4. pedidos 4](#_Toc203585464)

[5. usuarios 4](#_Toc203585465)

[6. reportes 4](#_Toc203585466)

[🧭 Relación entre apps 4](#_Toc203585467)

[💡 Recomendación técnica 5](#_Toc203585468)

## 🧱 Estructura de apps Django para Befine Web

| **App** | **Descripción** |
| --- | --- |
| core | Página principal, landing, contacto, ayuda, manual. Contiene contenido público. |
| productos | CRUD de productos: recargas, bidones, dispensadores, packs. Incluye precios, stock y categorías. |
| clientes | Registro, login, perfil del cliente, historial de pedidos, repetición de pedidos. |
| pedidos | Carrito, checkout, validación de zona y cantidad mínima, estado del pedido, historial y pagos. |
| usuarios | Administración de usuarios internos: dueños, administradores, repartidores. Gestión de roles. |
| reportes | Módulo exclusivo para el dueño/administrador, muestra ingresos diarios, historial de ventas, métricas clave. |

## 📦 Detalle de cada app

### 1. core

* Vistas: index, nosotros, contacto, soporte.html
* Lógica: solo contenido estático o semiestático
* Uso general y público

### 2. productos

* Modelo Producto: nombre, descripción, tipo, precio, stock, imagen, estado
* Categorías: Recarga, Recarga + Envase, Dispensador Básico, Pack Inicial, Dispensador Eléctrico
* Vistas: catálogo, detalle, administración

### 3. clientes

* Registro / login de clientes
* Historial de pedidos
* Repetir pedido
* Editar datos personales
* Validación por comuna

### 4. pedidos

* Modelo Pedido: cliente, productos, fecha, total, estado, método de pago
* Carrito, checkout, validación de zona y recargas mínimas
* Estados: "Pendiente", "En reparto", "Entregado", "Rechazado"
* Integración futura con WebPay

### 5. usuarios

* Modelo extendido de User con campo rol: dueño, admin, repartidor, cliente
* Panel de control: gestión de usuarios internos
* Login diferenciado por rol

### 6. reportes

* Vista exclusiva del dueño/admin
* Totales diarios de pedidos e ingresos
* Listado por fecha
* Dashboard básico (gráficas o tablas)

## 🧭 Relación entre apps

plaintext

CopiarEditar

core: muestra el inicio y contenido público

└── linkea a productos, login, contacto

productos: muestra catálogo al cliente

└── editable por admin o dueño

clientes: maneja usuarios finales registrados

└── hacen pedidos

└── pueden ver historial y repetir pedidos

pedidos: proceso de compra y validación

└── incluye carrito, resumen, pago

usuarios: administra personal interno

└── login y panel para dueño/admin/repartidor

reportes: módulo exclusivo del dueño/admin

└── muestra ventas, pedidos, usuarios

## 💡 Recomendación técnica

* Usa AbstractUser para extender el modelo y agregar el campo rol
* Organiza tus templates por app (templates/clientes/, etc.)
* Usa Bootstrap o Tailwind para tener una UI limpia y responsiva
* Para la validación por comuna, crea una tabla Zona o usa lógica directa en pedidos.views